Langley Research Center

SPIDER

DO-254 Case Study

Paul S. Miner

FAA National Software Conference

May 15, 2002

Langley Research Center

SPIDER

Outline

- Project Overview
  - Goals
  - Design Description
- Appendix B items
- Future Plans

May 15, 2002 DO-254 Case Study 2

Project Goals

- FAA Goals:
  - Develop case study application of DO-254
  - Provide feedback on problem areas
  - Provide material suitable for DO-254 training
- NASA Goals:
  - Demonstrate Application of Formal Methods in Certification context
  - Develop research platform for exploring recovery from correlated transient faults

Team Members and Responsibilities

- NASA
  - Paul Miner, Project Lead, Formal Modeling
  - Mahyar Malekpour, Design Engineer
  - Wilfredo Torres, Design Engineer
  - Kelly Hayhurst, Process Assurance
- ICASE
  - Alfons Geser, Formal Modeling, Independent Review

Project Overview

- Design part of a new fault-tolerant IMA architecture for case study
  - Fault-tolerance is inherently complex
  - but system description is compact
- Case study applied to the Reliable Optical Bus (ROBUS) of the Scalable Processor-Independent Design for EME Resilience (SPIDER).

What is SPIDER?

- A family of fault-tolerant IMA architectures
- Inspired by several earlier designs
  - Main concept inspired by Palumbo's Fault-tolerant processing system (U.S. Patent 5,533,188)
    - Developed as part of Fly-By-Light/Power-By-Wire project
  - Other ideas from Draper's FTPP, FTP, and FTMP; Allied-Signal's MAFT; SRI's SIFT; Kopetz's TTA; Honeywell's SAFEbus; . . .

Langley Research Center

SPIDER

SPIDER Architecture

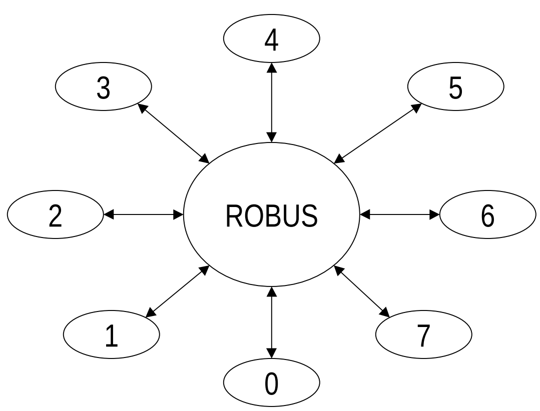
- *N* general purpose Processing Elements (PEs) logically connected via a Reliable Optical BUS (ROBUS)
- A ROBUS is an ultra-reliable unit providing basic fault-tolerant services
- A ROBUS is implemented as a special purpose fault-tolerant device
  - ROBUS contains no software

May 15, 2002 DO-254 Case Study 7

Langley Research Center

SPIDER

SPIDER Architecture



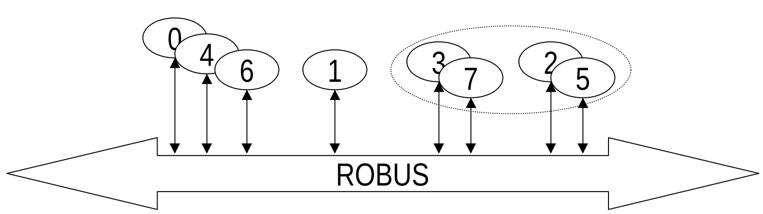
May 15, 2002 DO-254 Case Study 8

Logical View of ROBUS

- ROBUS operates as a time-division multiple access broadcast bus
- ROBUS strictly enforces write access
  - no babbling idiots (prevented by ROBUS topology)
- Processing nodes may be grouped to provide differing degrees of fault-tolerance
  - PEs cannot fail asymmetrically (prevented by ROBUS topology)

Logical view of ROBUS

(Sample Configuration)



SPIDER

ROBUS Characteristics

- Bus access schedule statically determined
  - similar to SAFEbus, TTA
  - All good nodes agree on schedule
- Some fault-tolerance functions provided by processing elements
  - ROBUS does not have general purpose processing capabilities
- Processing Elements need not be uniform
  - support for dissimilar architectures

SPIDER

ROBUS Requirements

1. All messages shall be broadcast on the ROBUS by the processing elements (PEs) according to a pre-determined message sequence. All good PEs shall agree upon the message sequence.

1.1 The ROBUS shall ensure the proper message sequence

1.1.1 A faulty PE shall not prevent a good PE from broadcasting in its allocated time slot

- No Babbling Idiots

Requirements (continued)

1.2 All fault-free PEs shall observe the exact same sequence of messages

1.2.1 If a faulty PE broadcasts a message, all good PEs shall agree on the content of the message.

1.2.2 If a good PE broadcasts a message, all good PEs shall receive the message that was broadcast.

- The ROBUS needs a Byzantine Fault Tolerant Interactive Consistency Protocol

Requirements (continued)

2. ROBUS shall provide a reliable time source (RTS) to all PEs

2.1 The ROBUS shall maintain synchronization in the presence of a bounded number of internal ROBUS component failures

2.2 All good PEs shall be synchronized relative to the ROBUS

- The ROBUS needs a Byzantine Fault Tolerant Clock Synchronization Protocol

Langley Research Center

SPIDER

Requirements (continued)

3. ROBUS shall provide correct and consistent system diagnostic information to all fault-free PEs in the presence of a bounded number of component failures.

4. ROBUS shall be an order of magnitude more reliable than is required for the supported aircraft function.

4.1 (Level A) For 10 hour mission, P(Failure) < 10^{-10}

May 15, 2002 DO-254 Case Study 15

Langley Research Center

SPIDER

Design Assurance Strategy

- Fault-tolerance protocols and reliability models use the same fault classifications
- Reliability analysis using SURE (Butler)
  - Calculates P(enough good hardware)
- Formal proof of fault-tolerance protocols using PVS (SRI)

enough good hardware => correct operation

May 15, 2002 DO-254 Case Study 16

Langley Research Center

SPIDER

Physical Segregation

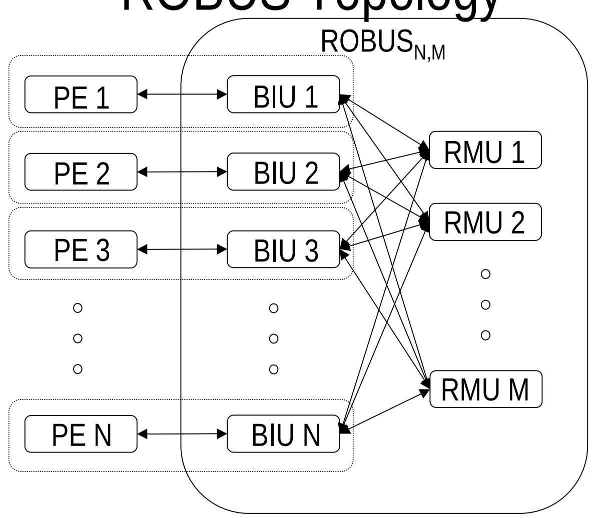
- ROBUS decomposed into physically isolated Fault Containment Regions (FCR)
  - Two main design elements
    - Bus Interface Unit (BIU)
    - Redundancy Management Unit (RMU)
  - Processing elements may form separate FCRs
- FCRs fail independently

May 15, 2002 DO-254 Case Study 17

Langley Research Center

SPIDER

ROBUS Topology



May 15, 2002 DO-254 Case Study 18

Langley Research Center

SPIDER

Fault Assumptions

- The failure status of an FCR is subdivided into four cases
  - Good (or fault-free)
  - Benign faulty (Obviously bad to all good)
  - Symmetric Faulty (Same manifestation to all good)
  - Asymmetric Faulty (Byzantine)
- Models use these classifications
- This is a global classification

May 15, 2002 DO-254 Case Study 19

Langley Research Center

SPIDER

Local Fault Classification

- Hybrid fault model implies ability to locally detect and diagnose all benign faulty nodes
- Each node maintains a local determination of fault status of other nodes
  - No good node is accused by any good observer
  - All benign faulty nodes are accused by all good observers
  - If a symmetric faulty node is accused by any good observer, then it is accused by all good observers
  - Asymmetric faulty nodes may be accused by some good observers

May 15, 2002 DO-254 Case Study 20

Maximum Fault Assumption

1. |GB| > |AB| + |SB|
2. |GR| > |AR| + |SR|
3. |AR| = 0 or |AB| = 0

All protocols to be verified under this fault assumption

Reliability model failure conditions correspond to violations of these assumptions

Outline

- Project Overview
  - Goals
  - Design Description
- **Appendix B items**
- Future Plans

Appendix B Items

- Architectural Mitigation
- Product Service Experience
- Advanced Verification Methods
  - Elemental Analysis
  - Safety-Specific Analysis
  - Formal Methods

Not relevant to this design

- Architectural mitigation
  - The ROBUS is an architecture designed to mitigate effects of various faults, so we cannot use as a strategy for its design assurance
- Service History - New design, so N/A
- Safety-specific analysis - This design is independent of aircraft function, so N/A

Elemental Analysis

- DO-254 analog of structural coverage
- Selected TransEDA's VN-cover tool for coverage analysis
  - Supports several different types of coverage
  - Control logic tests
    - statement, branch, condition, path
  - Data tests
    - trigger, signal trace, toggle

Focused Expression Coverage

- VN-cover's default condition coverage for VHDL code is Focused Expression Coverage (FEC)
- We have determined that FEC is the same as Masking MC/DC
  - By examining TransEDA documentation
  - By analyzing results for simple designs

Assessment of VN-cover

- DO-254 does not require detailed assessment of tools supporting elemental analysis
  - *"If the tool is … used to assess the completion of verification testing, such as in elemental analysis, no further assessment is necessary"* p. 76, item 4.

Planned uses of VN-cover

- FEC for both BIU and RMU
- Explore other coverage measures such as toggle and trigger

Langley Research Center

SPIDER

Formal Methods

- This is dominant design assurance strategy for this project
- Emphasis on early life-cycle verification
- Formal proof of key fault-tolerance protocols
  - Interactive Consistency
  - Distributed Diagnosis
  - Clock Synchronization

May 15, 2002 DO-254 Case Study 29

Langley Research Center

SPIDER

Strength of Formal Verification

- Proofs equivalent to testing the protocols
  - for all possible ROBUS configurations (i.e. for all N, M)
  - for all possible combinations of faults that satisfy the maximum fault assumption for each possible ROBUS configuration
  - for all possible message values
- The PVS proofs provides verification coverage equivalent to an infinite number of test cases.
  - Provided that the PVS model of the protocols is faithful to the VHDL model

May 15, 2002 DO-254 Case Study 30

Interactive Consistency

(Byzantine Agreement)

Agreement: For any message, all good receiving nodes will agree on the value of the message

Validity: If the originator of the message is non-faulty, good receivers will receive the message sent

Diagnosis

Correctness: Every node diagnosed as faulty by a good node is faulty

- A good node can never conclude that another good node is faulty

Completeness: Every faulty node is (eventually) diagnosed as being faulty

- This is not always possible (pathological case involves asymmetric fault)

- Also need Agreement among good nodes

Langley Research Center — SPIDER

Clock Synchronization

Precision: There is a small positive constant d_{max} such that for any two clocks that are *good* at *t,*

$|C_1(t) - C_2(t)|$ ☼ d_{max}

Accuracy: All good clocks maintain an accurate measure of the passage of time (within a linear envelope of real time)

May 15, 2002 DO-254 Case Study 33

Langley Research Center — SPIDER

Interdependencies

- Each of these protocols depends upon the correct operation of the others
  - The IC and Diagnosis protocols are synchronous distributed algorithms, they require the relative skew between any pair of good nodes be bounded
  - All protocols depend upon correct diagnostic data for ignoring failed nodes (This uses a combination of Local and Global Diagnosis)
  - Global diagnosis protocol uses Interactive Consistency for exchange of local error syndromes

May 15, 2002 DO-254 Case Study 34

SPIDER

Discussion of Protocols

- Overview of Interactive Consistency Protocol
- Model characteristics
- What to look for in formal models

SPIDER

Interactive Consistency

- SPIDER IC protocol is simple adaptation of IC algorithm for Draper FTP Architecture
  - Existing PVS proof (for FTP) due to Lincoln and Rushby, *COMPASS'94*, pages 107-120
  - SPIDER Protocol is similar to the original FTP protocol [T. Basil Smith, FTCS 14 (1984)]
- Protocol generalizes one suggested in

  Daniel Davies and John Wakerly, Synchronization and Matching in Redundant Systems, *IEEE Trans. on Computers*, Vol. C-27, No. 6, June 1978

Langley Research Center — SPIDER

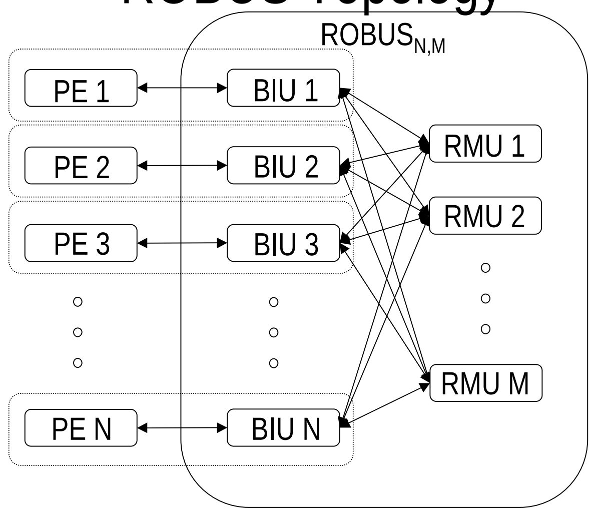
Interactive Consistency Protocol (ICP)

1. PE *j* transmits its message *v* to BIU *j*
2. BIU *j* broadcasts *v* to all RMUs
3. For each RMU *k*, if RMU *k* does not receive a correctly formatted message from BIU *j* then it broadcasts *source error* to all BIUs, otherwise it broadcasts the received value v_k to all BIUs
4. Each BIU collects the values received (v_1 , …, v_M). .If a BIU does not receive a correctly formatted message from RMU *k*, it removes RMU *k* from its set of *trusted* RMUs (*k* is accused).
5. Each BIU determines if there is a majority among the values from the *trusted* RMUs
6. If BIU *l* determines that a majority of *trusted* RMUs sent the same value v_{maj}, BIU *l* transmits v_{maj} to PE *l*. Otherwise, BIU *l* transmits *no majority* to PE *l*.

May 15, 2002 DO-254 Case Study 37

Langley Research Center — SPIDER

ROBUS Topology



May 15, 2002 DO-254 Case Study 38

Langley Research Center

SPIDER

PVS Model of IC Protocol

- Global view of protocol (local information modeled using global vectors)
- Assumes synchronous composition
- Communication primitive modeled using full knowledge of global fault status. Behavior of faulty nodes is only restricted by global fault status and communication interface.
- Vote using updated set of trusted sources based on local diagnosis modeled in the communication primitive

May 15, 2002 DO-254 Case Study 39

Langley Research Center

SPIDER

Interactive Consistency Results(1)

Agreement: For all BIU *g*,

if (|AR| = 0) or

(g ♒ AB and |GR| > |SR| + |AR|),

then for all *p,q* ♑ GB:

ICP(g,v,p) = ICP(g,v,q)

May 15, 2002 DO-254 Case Study 40

Langley Research Center

SPIDER

Interactive Consistency Results(2)

Validity:

If |GR| > |SR| + |AR|, then for *p* ♑ GB :

- If *g* ♑ GB, then *ICP(g,v,p) = v*
- If *g* ♑ BB, then *ICP(g,v,p) = source error*
- If *g* ♑ SB, then *ICP(g,v,p) = sent(g,v)*

May 15, 2002 DO-254 Case Study 41

Langley Research Center

SPIDER

Critical Assumptions of IC

- Nodes are synchronized within a bounded skew and architecture prevents this skew from impacting operation of protocol
- Local diagnostic information is correct
  - Sources for vote by a good node include all good nodes, no benign faulty nodes, and only those symmetrically faulty nodes included by all other good nodes
  - Benign faults are excluded by local diagnosis
- Voter has required properties
  - Have PVS proof of Boyer-Moore MJRTY algorithm
- Communication primitives have required properties

May 15, 2002 DO-254 Case Study 42

Langley Research Center

SPIDER

Modeling Issues

- Are the models meaningful?
  - Are abstractions valid?
    - e.g. synchronous composition, functional abstraction
  - Are assumptions satisfiable?
    - Is there a typical case?
    - Are assumptions true for initial conditions?
    - Are assumptions preserved through execution of protocol?

May 15, 2002 DO-254 Case Study 43

Langley Research Center

SPIDER

More Modeling Issues

- How is the formal model related to the modeled artifact?
  - Compilation of VHDL to model?
  - Compilation of model to VHDL?
  - Manual comparison?

May 15, 2002 DO-254 Case Study 44

Langley Research Center

SPIDER

Formal Proof Issues

- Have you proven the claim you intended to prove?
  - Sanity checks:
    - For each hypothesis, demonstrate why proof fails when hypothesis removed (may be an informal argument)
    - Confirm that you haven't assumed the conclusion
    - Confirm that models of system components only have access to data that the modeled component has access to.

May 15, 2002 DO-254 Case Study 45

Langley Research Center

SPIDER

Added Benefits of Formal Methods

- Formal Models provide detailed understanding of why protocols work
- This sometimes results in ability to recognize improvements to protocols
  - verification of diagnosis protocol suggested way to reduce communication overhead by almost half
  - subsequently identified more aggressive optimization
    - currently verifying new protocol

May 15, 2002 DO-254 Case Study 46

Langley Research Center

SPIDER

Future Plans

- Complete verification data
  - VHDL test benches
  - Coverage analysis using VN-cover
  - complete formal proofs
- Revise design to incorporate transient fault recovery
- Update FPGA based lab prototype

May 15, 2002 DO-254 Case Study 47